



# Visual Basicによるシンクライアント アプリケーションの構築

---

株式会社SASインスティテュートジャパン  
プロフェッショナルサービスグループ第2部  
村山友子  
段谷高章

# 本日の内容

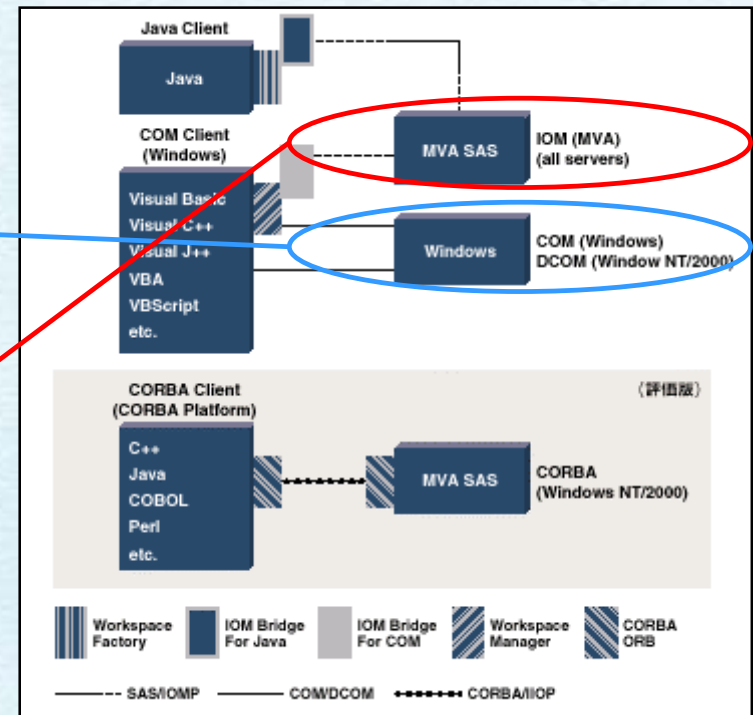
- SAS Integration Technologies
- SASセッションの開始
- SASプログラムの実行
- 結果の取得
- サンプルアプリケーションの紹介

# 1. SAS Integration Technologies

- IOM(Integrated Object Model)
  - 分散オブジェクトとして利用可能なインターフェイスを提供
  - 業界標準規格とのシームレスな統合が可能
  - 様々な開発言語を用いた開発が容易

- 接続プロトコル

- Windows環境
  - COM/DCOM
  - IOM (SAS独自のプロトコル)
- その他の環境
  - IOM (SAS独自のプロトコル)



# 1 . SAS Integration Technologies

## COM/DCOMとIOMの比較

### ■ COM/DCOM

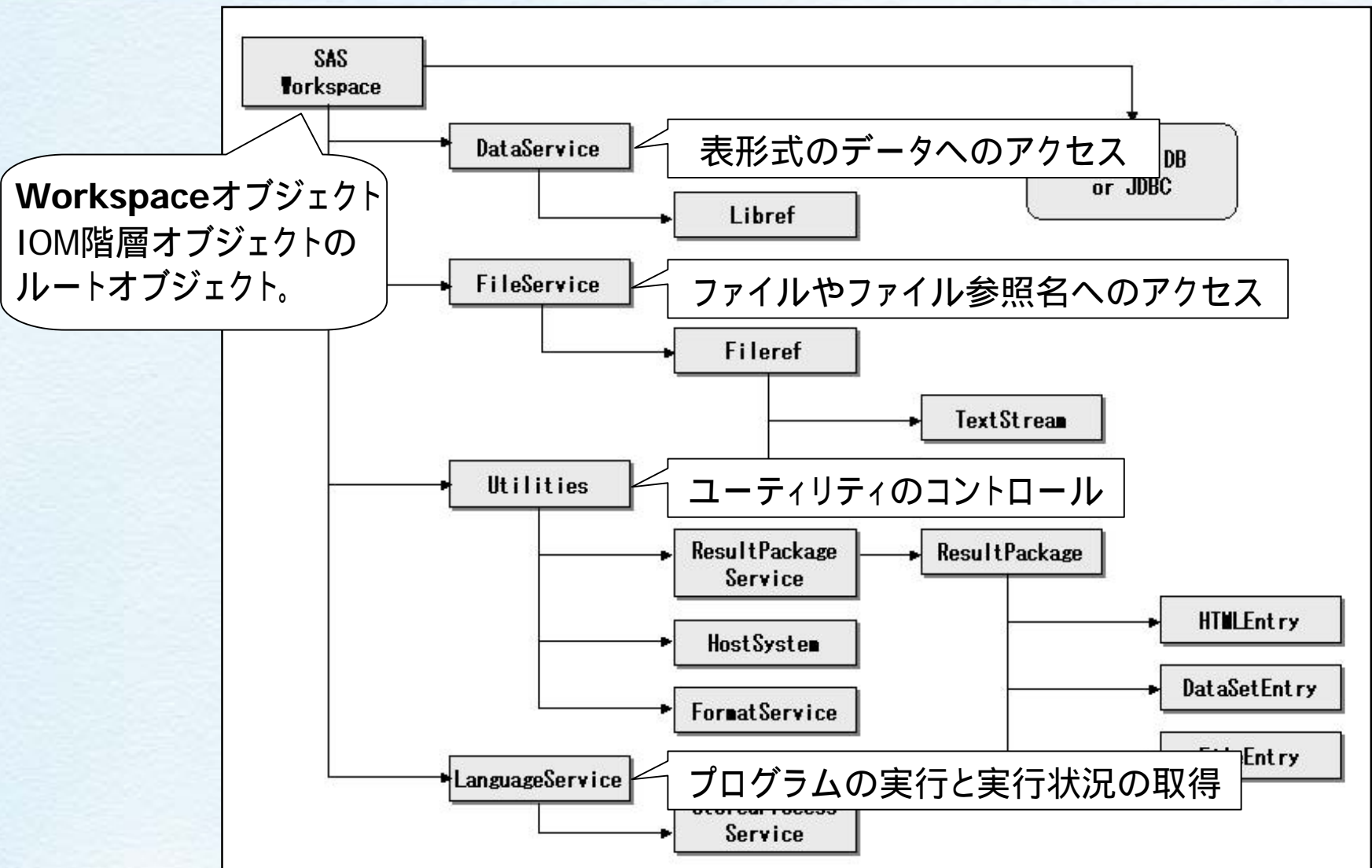
- ADO(ActiveX Data Object)
- OLE DB(Object Linked Embedding for Databases)
- Windows環境の標準プロトコルを使用
- DCOMCNFGによる一元管理
- Windows標準のSCMサービス

### ■ IOM

- MVAのサポート & SASサーバー設定の複数保持が可能
- スポンサー設定ファイルによる管理
- オブジェクトスポーナ



## 2. SASセッションの開始



## 2 . SASセッションの開始

- Workspaceオブジェクト(ルートオブジェクト)の作成
  - 接続プロトコルによって必要な情報が異なる

### DCOMの場合

```
obServerDef.Protocol = ProtocolCom    ' 接続プロトコルを指定
obServerDef.Name = "DCOMConnection"    ' 接続定義の名前
obServerDef.MachineDNSName = g_strMachineName    ' 接続先のマシン名
Set obWS = obWSMgr.Workspaces.CreateWorkspaceByServer("My workspace", _
    VisibilityProcess, obServerDef, "", "", strError)
```

### IOMの場合は上記以外に下記の内容が必要

```
strUserID = "MyNTUserName"    ' ユーザ名/パスワードを指定
strPassword = "MyPassword"
obServerDef.Port = 5307    ' ポート番号を指定
Set obWS = obWSMgr.Workspaces.CreateWorkspaceByServer("My workspace", _
    VisibilityProcess, obServerDef, strUserID, strPassword, strError)
```

# 3 . SASプログラムの実行

SASサーバー上でプログラムを実行する方法は2つ

- LanguageService.Submitメソッドの利用
  - SASプログラムをVBコード中に記述する方法


obSAS.LanguageService.Submit \_

“data a; do x=1 to 10; y=x\*x\*x;” & \_

“output; end; run;”

# 3 . SASプログラムの実行

- StoredProcessServiceの利用
  - サーバー上のSASプログラムを実行
  - クライアントからパラメータを取得
  - VBプログラムとSASプログラムの分離

 メンテナンスが容易

```
Dim obStoredProcessService As SAS.StoredProcessService
```

```
Set obStoredProcessService =  
obSAS.LanguageService.StoredProcessService
```

‘ ストアドプログラムの保存場所の指定と実行

```
obStoredProcessService.Repository = "file:c:¥temp"
```

```
obStoredProcessService.Execute "looper", "loopTimes=10"
```



# 3 . SASプログラムの実行

## ■ Executeメソッドの実際とストアドプロセス

### ■ Executeメソッド

obStoredProcessService.Execute "looper", "loopTimes=" & LoopNum & "

### ■ ストアドプロセス

```
%let loopTimes= ;  
*ProcessBody;
```

□ プロローグセクション

```
data a;  
  do x=1 to &looptimes;  
    y=x*x*x;  
    output;  
  end;  
run;
```

□ プログラム本文

## 4. 結果の取得

### ■ Logの取得

#### ■ StepErrorメソッド (LanguageServiceオブジェクト)

クライアントがWindows2000という環境において、この機能を使用するためには、Windowsに付属しているdcomcnfgプログラムを使用してセキュリティに関する設定の変更が必要

### ■ Outputの取得

#### ■ 外部ファイルに出力した結果を取得

ODSの機能を利用して出力した結果をVBで表示するように作成することで、結果を画面に表示することが可能

#### ■ SASデータセットの表示

結果をデータセットに保存し、ADOの機能を利用して、SASデータセットをDataGridに表示することが可能